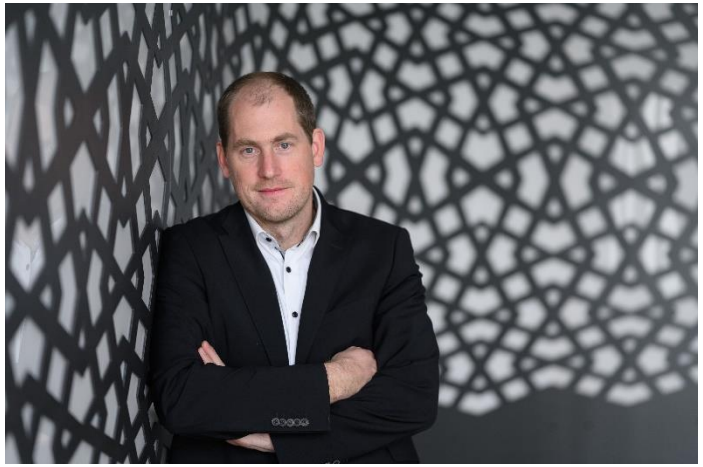
SCRUM Product Owner Schulung

Florian Eckkrammer



- *Seit 2006 am Technikum Wien (FH und Academy)*
- *Vorher/parallel tätig im Software Engineering im Bereich Smart Homes, technische Informatik, eHealth, und Entwicklung von LOB Applikationen*

- *seit Jänner 2021 Geschäftsführer der FH Technikum Wien*
- *2018-2020 Leiter des Departments Informatik*
- *Lehrgangsleiter App- sowie Web Development*
- *Stv. Studiengangsleiter Wirtschaftsinformatik*

- *Lehre schwerpunktmäßig in den Bereichen*
  - *Software Engineering*
  - *Development*
  - *Agile Methoden & Agile Transformation*
  - *System Integration*

- *Beratung von Firmen in Bereichen des SWE und der Agile Transformation*

- *Mitveröffentlichung von „Handbuch IT-Projektmanagement: Vorgehensmodelle, Managementinstrumente, Good Practices“*

Inhalte

- ✓ *Grundlagen Agilität*
- ✓ *Scrum als Vorgehensmodell*
- ✓ *User Stories*
- ✓ *Story Map*
- ✓ *Scrum Skalierung*

Warum Agile?

Oder der Umgang mit Veränderungen im Projekt

Ist Ihr Entwicklungsprozess ein: definierter oder ein empirischer Prozess?

definierter Prozess?

- präzise beschreibbar
- Anfangsbedingungen bekannt
- maximal kompliziert aber beherrschbar

??

…

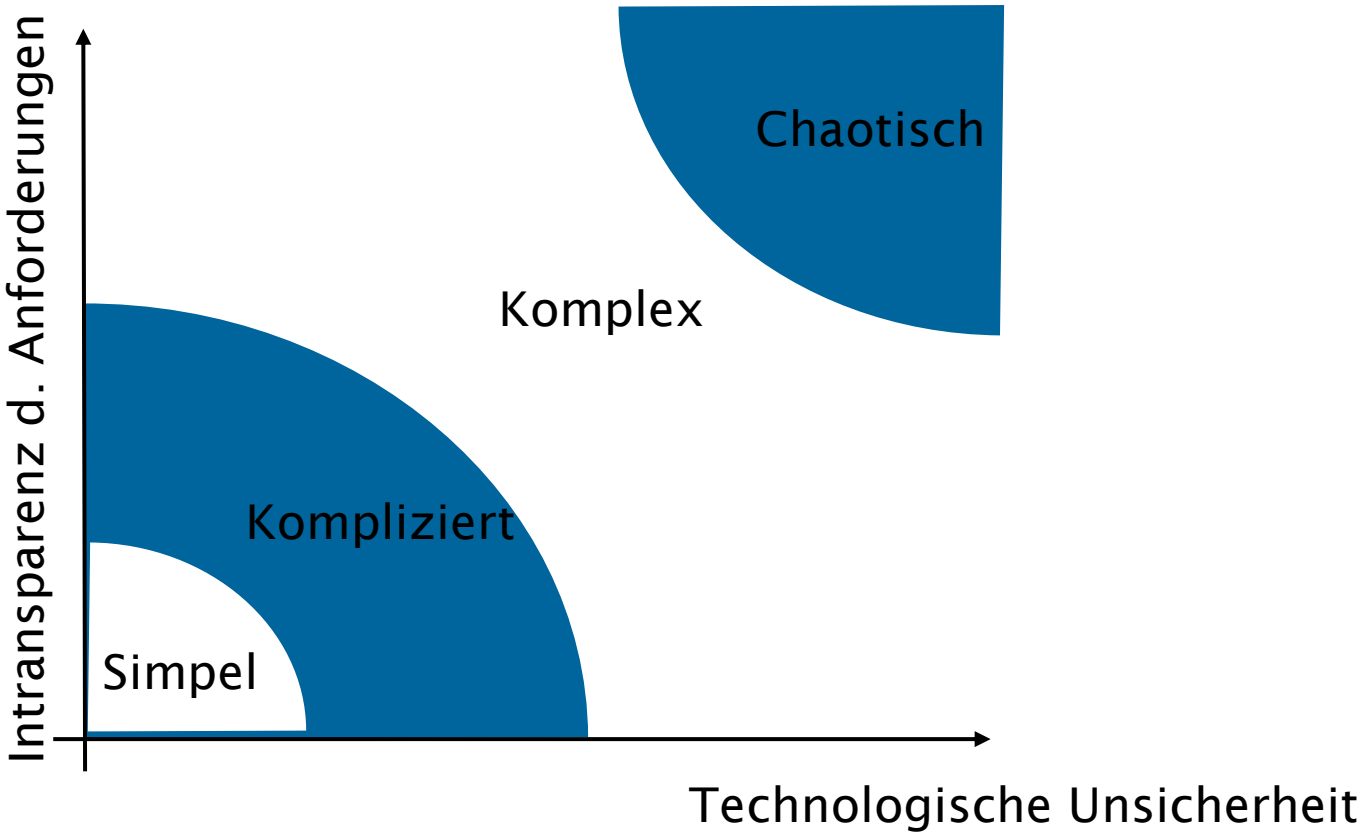
Oder doch eher ein empirischer Prozess?

- Umgebung und Anfangsbedingungen nicht vollständig bekannt
- Variable Anforderungen in Abhängigkeit der Zeit
- Es gibt kein „bestes Vorgehen“

??

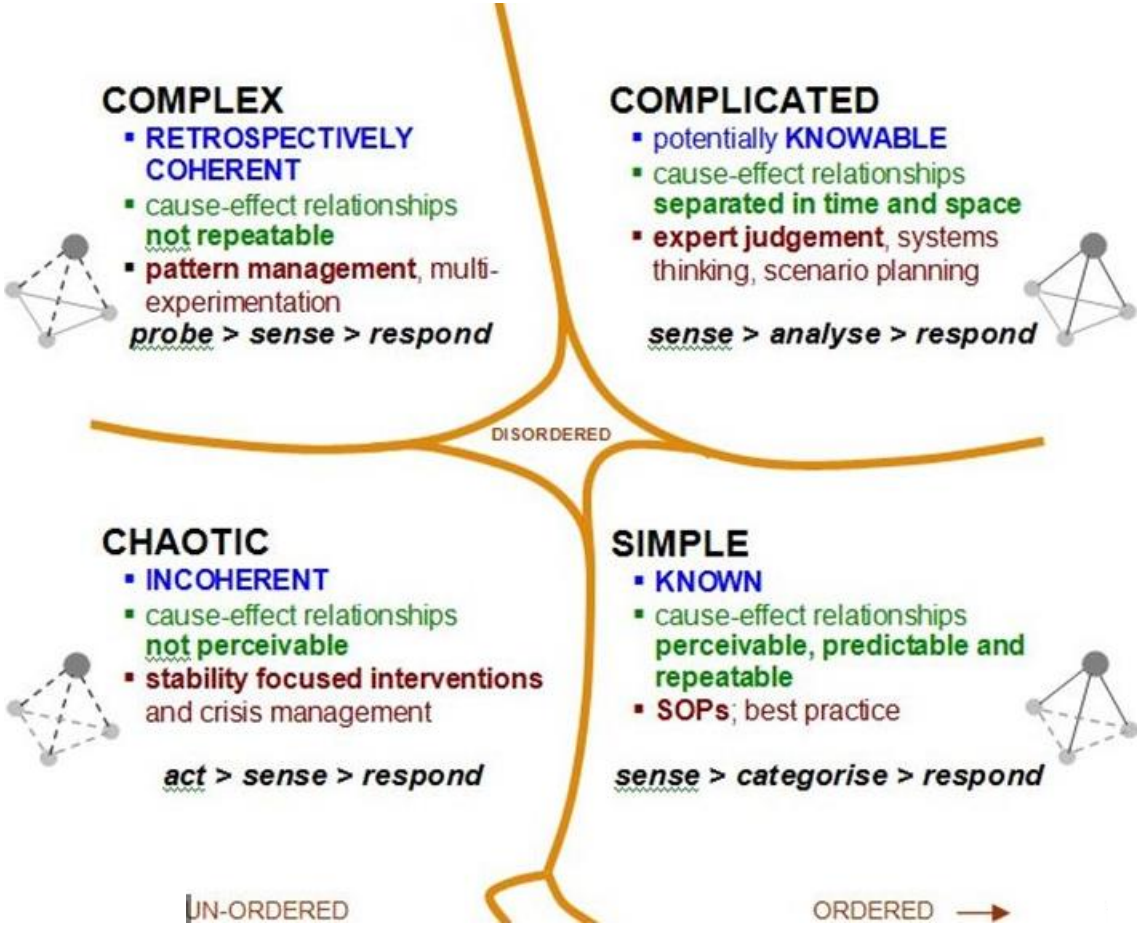
Wo liegt der Unterschied zwischen Komplex und kompliziert

https://pixabay.com/



Quelle: nach R. Stacey & K. Schwaber

5 Domänen: complex, complicated, chaotic, simple, disordered



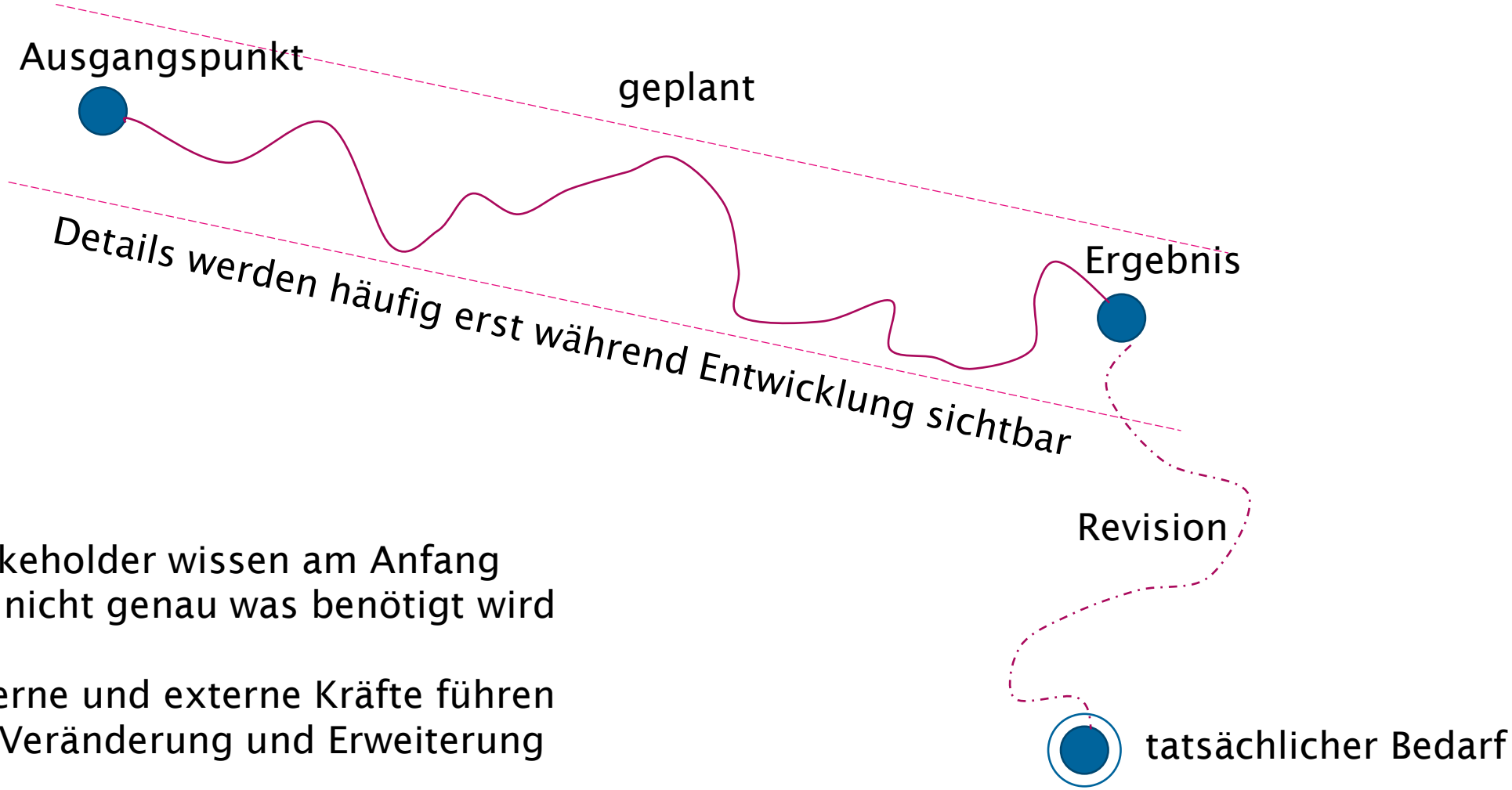
Cynefin Framework - Nach Dave Snowden 2002

Ergo:

Komplexe Projekte mit statischen Methoden, die nicht auf Änderungen reagieren können, **durchzuführen ist problematisch**.

Aber:

Was macht unsere Projekte denn so Komplex?

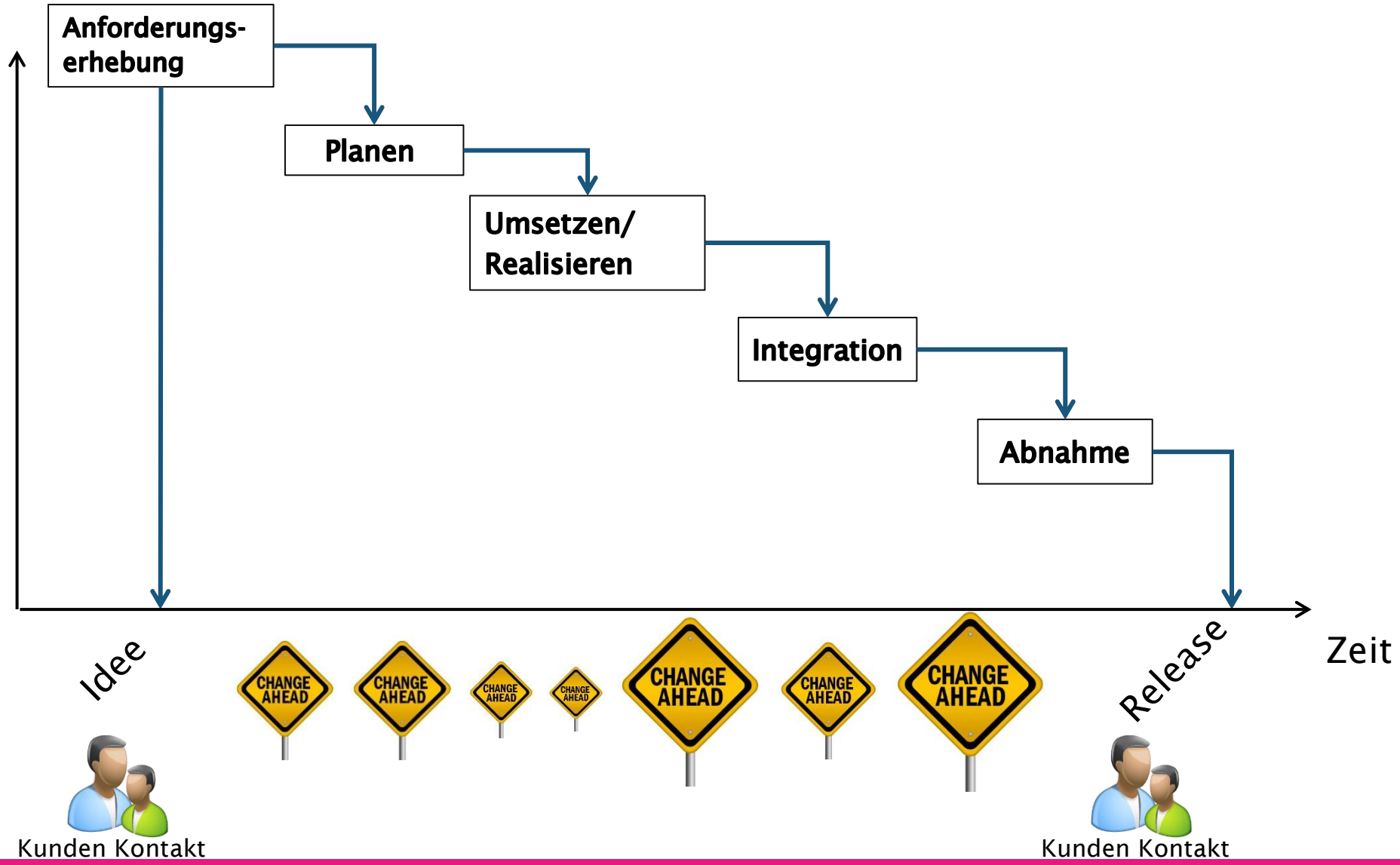


Stakeholder wissen am Anfang
oft nicht genau was benötigt wird

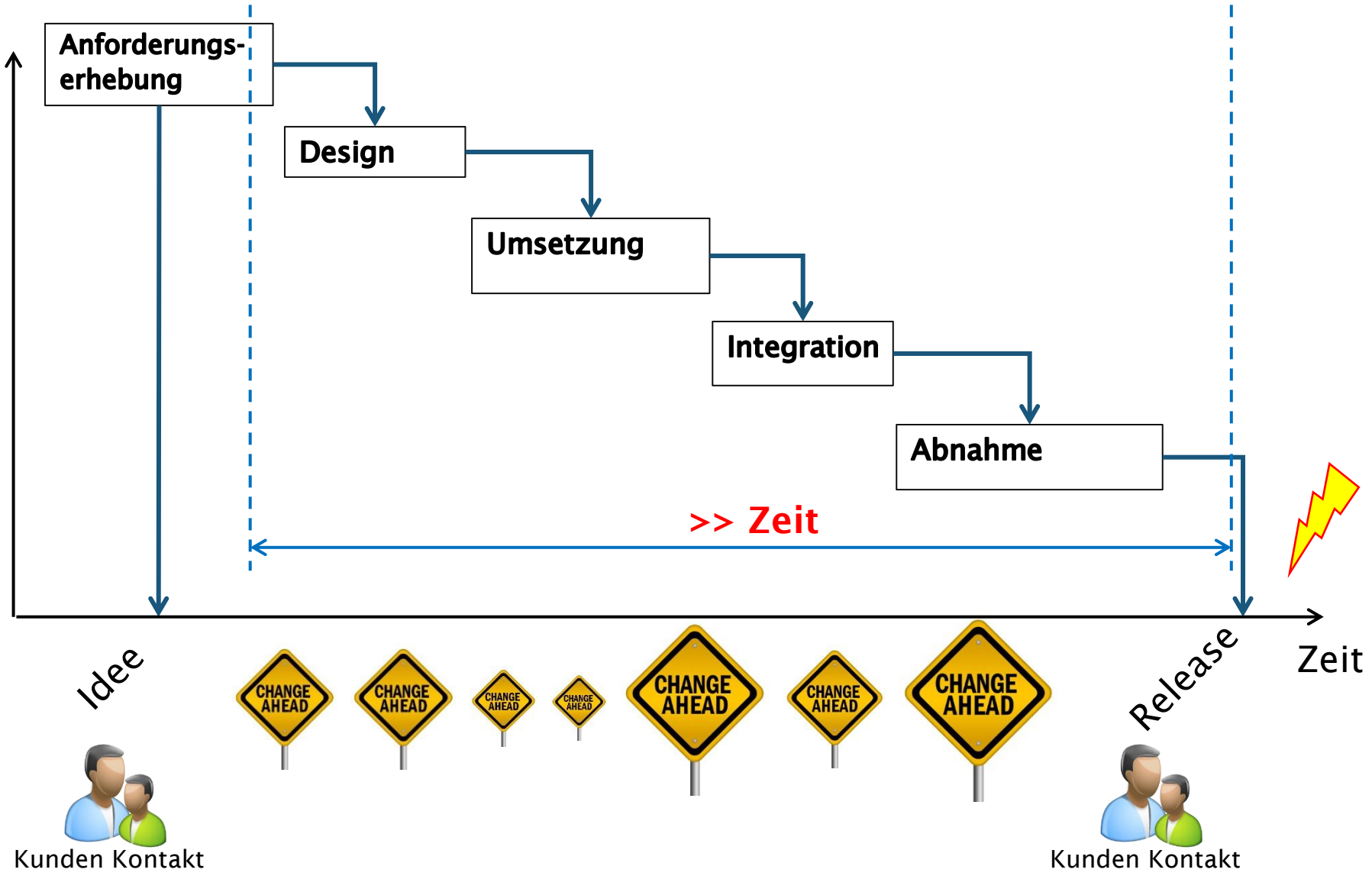
Interne und externe Kräfte führen
Zu Veränderung und Erweiterung

Quelle: basierend auf Mitch Lacey 2011

Veränderungen im Projekt



Veränderungen im Projekt



Problem

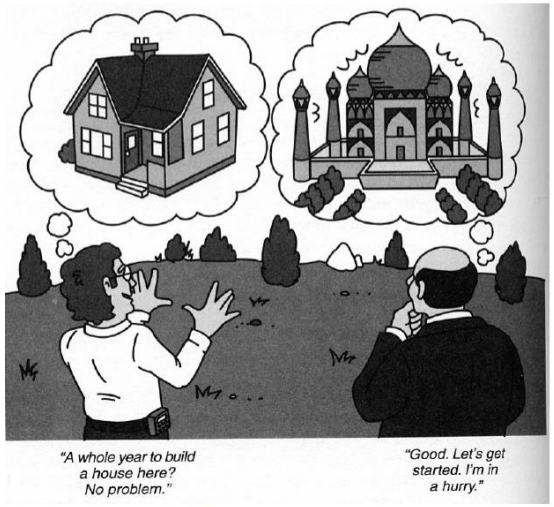
Kunde wird enttäuscht sein

Warum?

Problem

1) *Kunde kann Projektergebnis nicht einsetzen weil es nicht (mehr) den Anforderungen des Prozesses entspricht.*

2) *Kunde hat sich etwas anderes vorgestellt.*
   - *Oft sind Projektgegenstände ist abstrakt*
   - *Erwartungen wurden nicht erfüllt*



Source: Steve McConnell

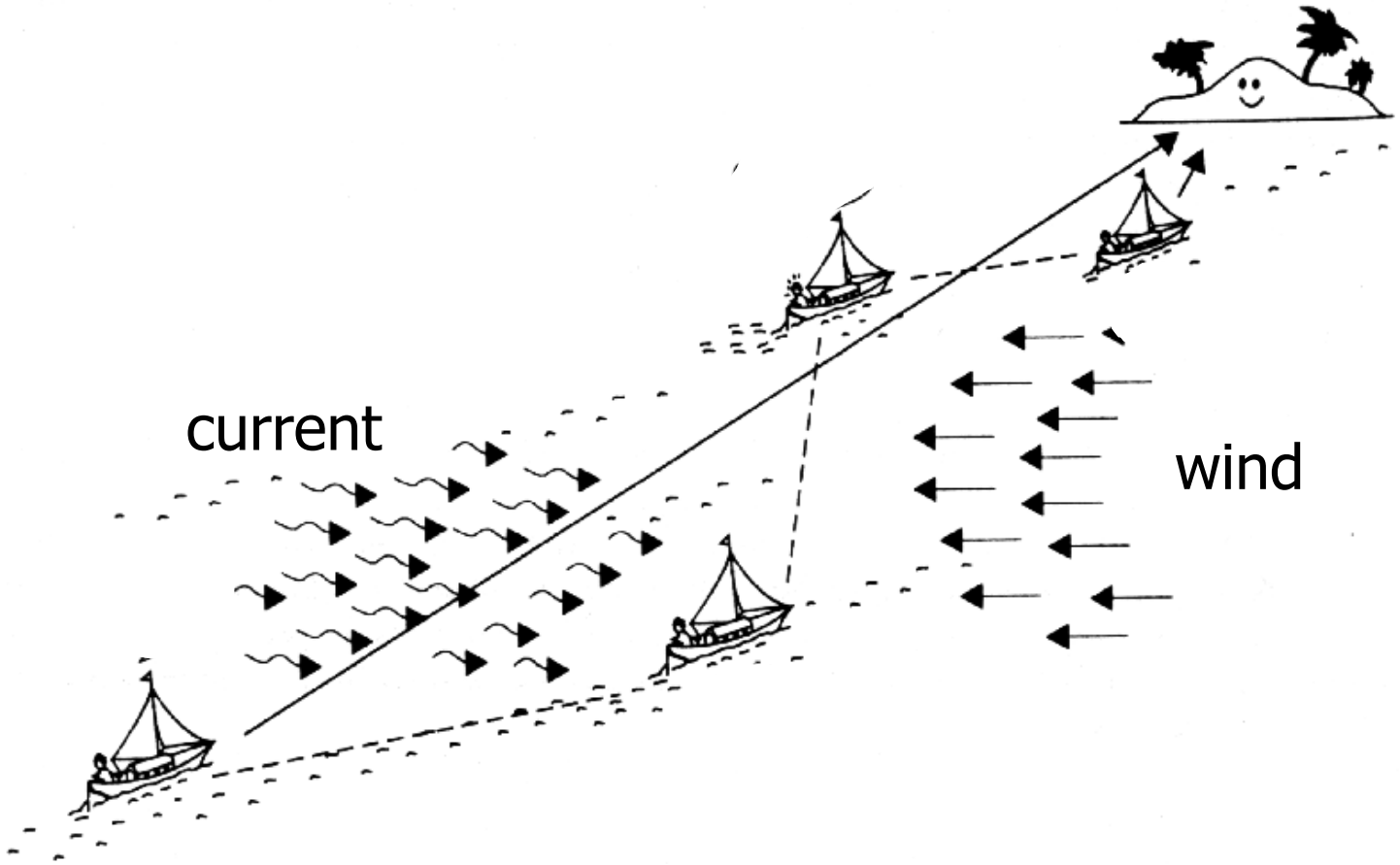
Problem

Unerfüllte Erwartungen sind problematisch

Vertrauen entsteht durch Vorhersehbarkeit
(Output entspricht Erwartungen)

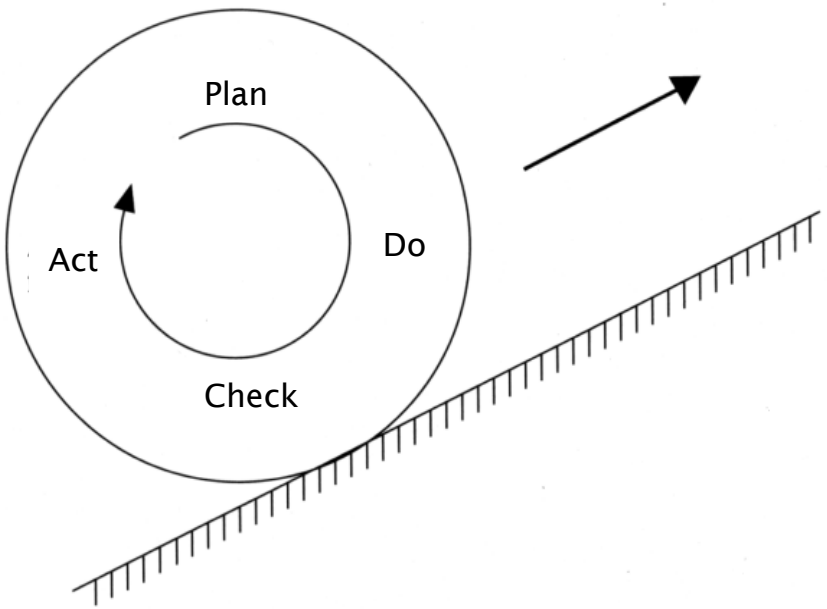
Handlungsspielraum des Kunden bei Projektende == null.

Regelkreis



Regelkreis

Plan - Do - Check – Act (nach Deming)



Plan: Planungsaufgaben, Vision, PSP, PAP,…

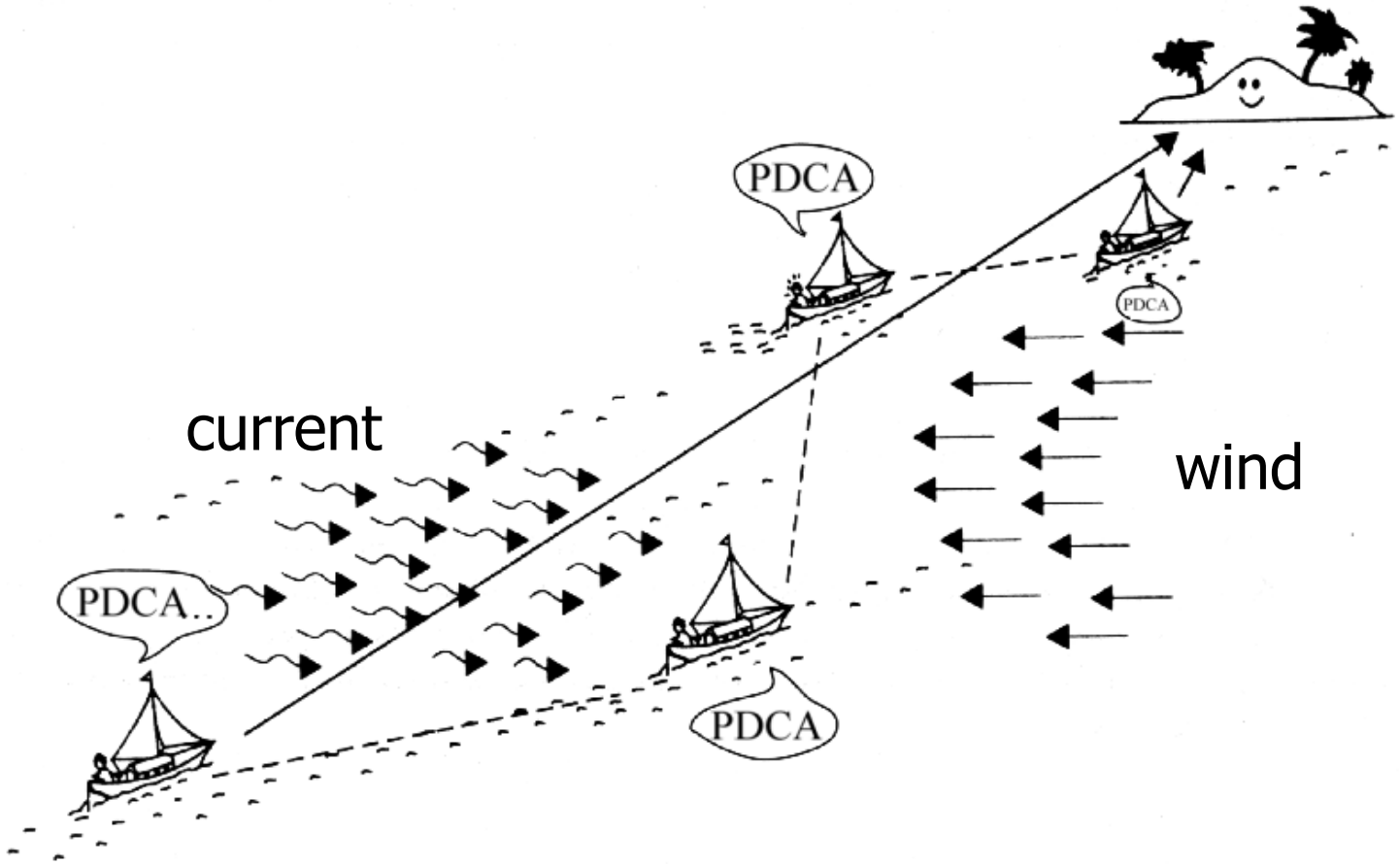
Do: Umsetzung (das Ausführen des geplanten)

Check: Überprüfen ob die Pläne korrekt implementiert und erreicht werden

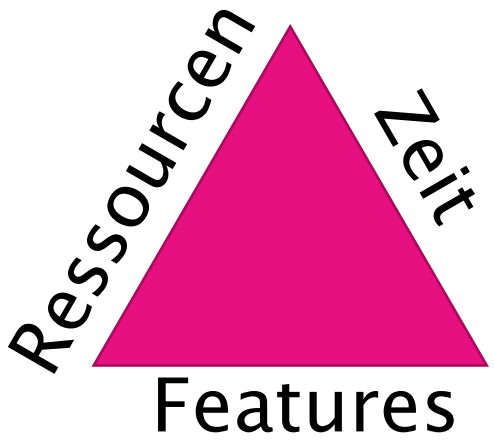
Act: Falls Ziele nicht erreicht werden analysieren, Maßnahmen entwickeln und anwenden

PDCA ist ein Phasenorientierter und zyklischer Prozess

Regelkreis

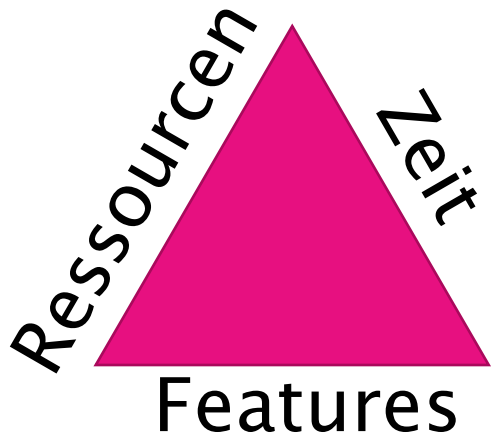


Projektmatrix bei traditionellen Projekten



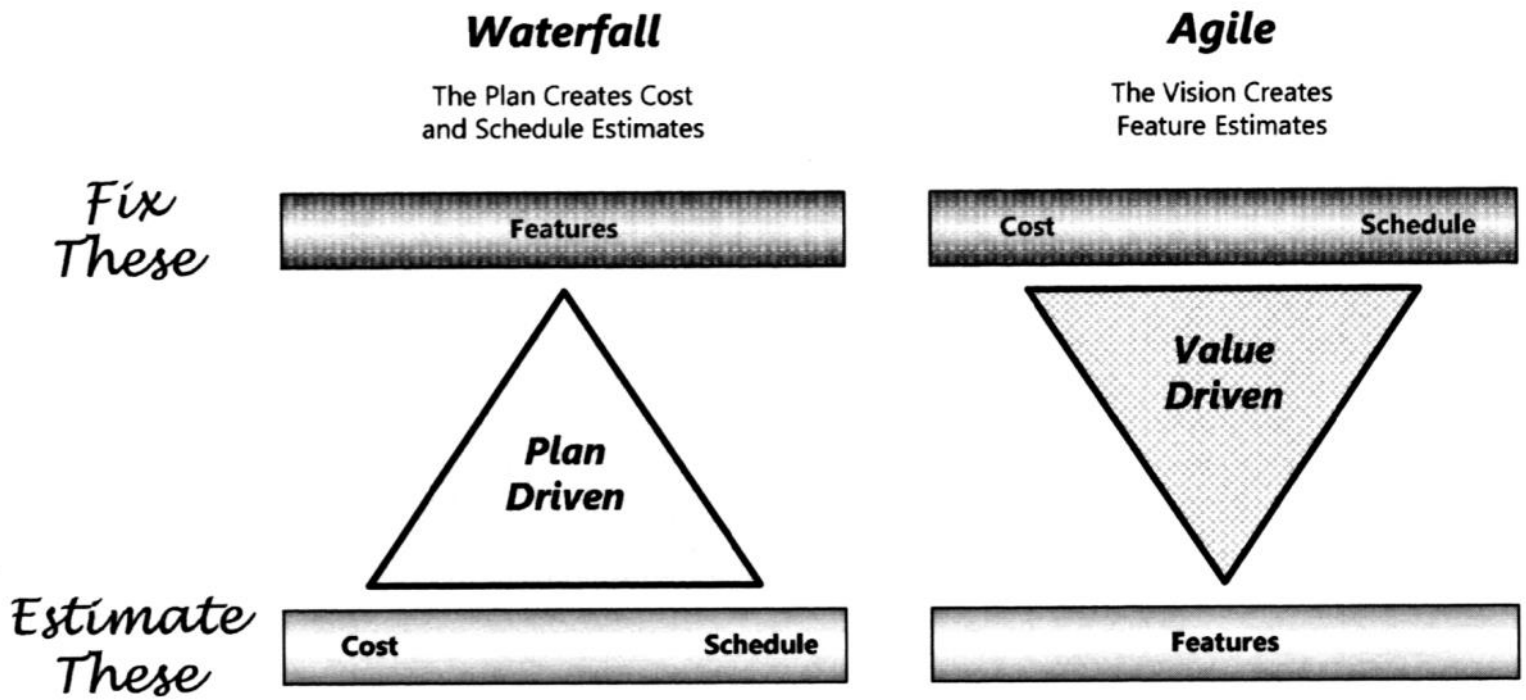
	geschätzt	Fix
Ressourcen	✓	
Zeit	✓	
Features		✓

Projektmatrix bei iterativen Projekten



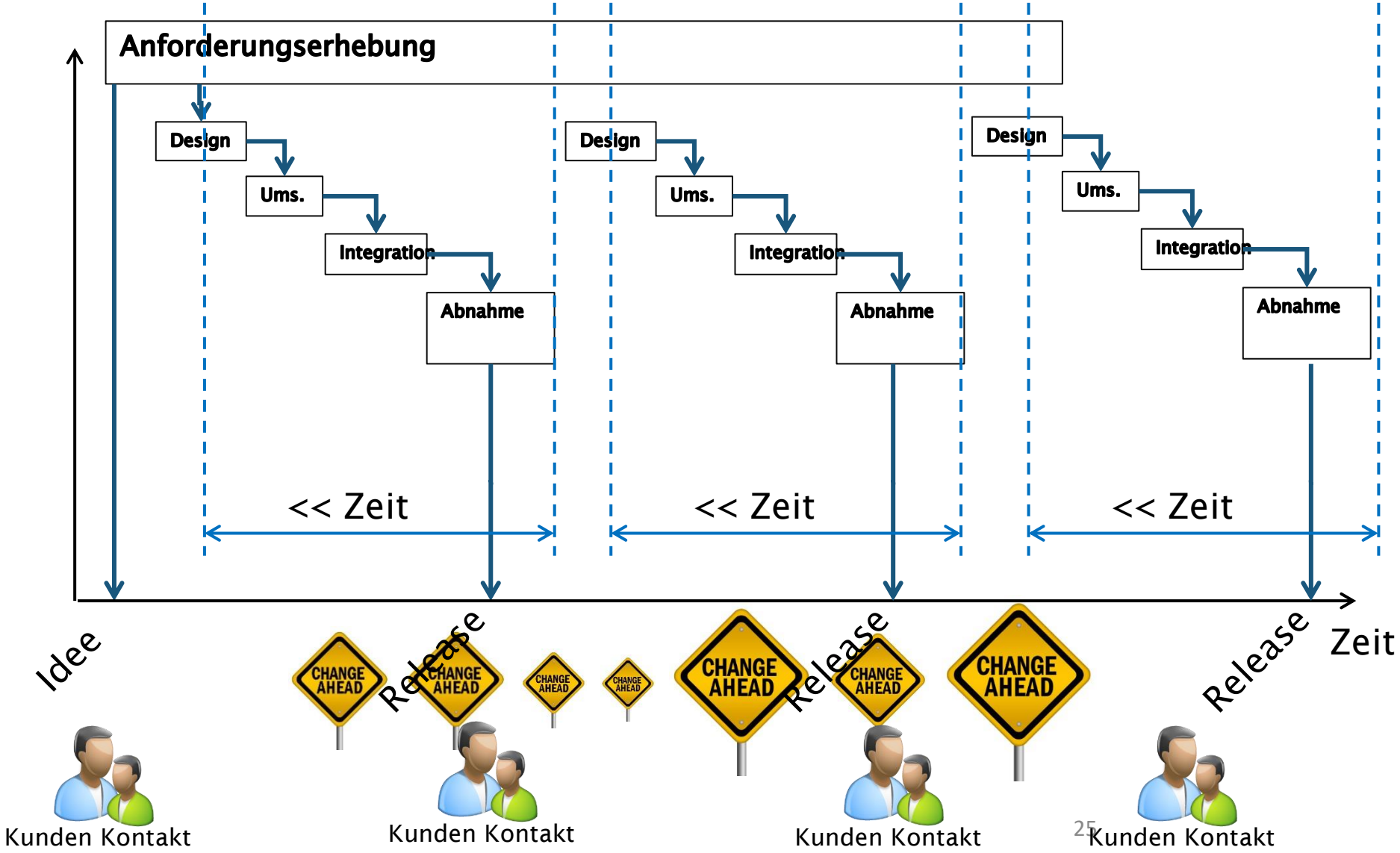
	geschätzt	Fix
Ressourcen		✓
Zeit		✓
Features	✓	

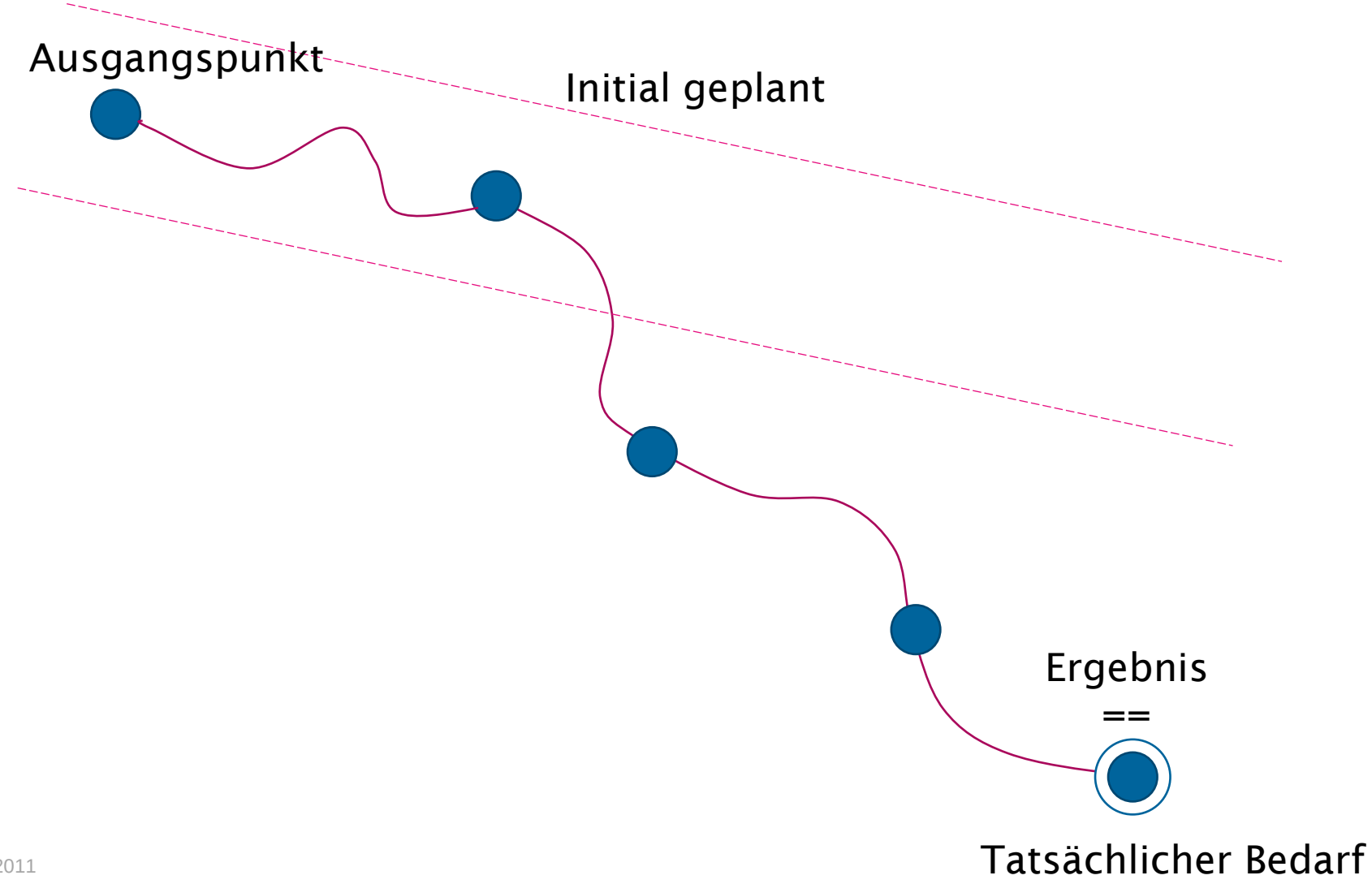
Gegenüberstellung – Traditionell vs. Agil



Source: Mitch Lacey 2011

Iterativer Ansatz

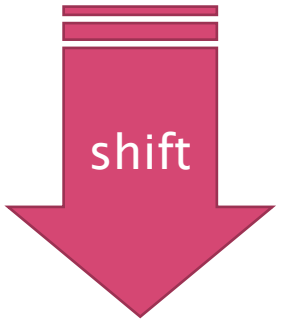




Quelle: basierend auf Mitch Lacey 2011

Vorgehensmodelle

Sequentielle Ansätze
- Wasserfall model
- V-Model



Iterative Ansätze
- Iteratives Phasen model
- Spiral model
- Explorative Prototyping model
- Evolutionary Software development model

Agile Ansätze …

Iterative vs. Agile

Iterative Ansätze

- Iteratives Phasen model
- Spiral model
- Explorative Prototyping model
- Evolutionary Software development model
- PDCA
- …

Agile Ansätze

- XP
- SCRUM
- Crystal Family
- FDD
- …

Iterative vs. Agile

Agile Vorgehensmodelle sind (hoch) iterativ

Unterschied liegt im Mindset und der Team-/Unternehmenskultur.

Pause

Verschiedene Ansätze

Es gibt sehr viele verschiedene Methoden und Ansätze.

Unterschiede in:

- Gültigkeit,
- Abstraktion
- Verbreitung
- Anwendungsbezug
- …

Übersicht

Zwei Interessante Bereiche im relevanten Kontext:

- Lean
  - Lean Production
  - Lean Startup
  - Lean Management
  - Lean whatever…

- Agile
  - Agile manufacturing
  - agile production
  - Agile systems engineering
  - Agile software engineering
-

Lean

Lean im allgemeinen geht es darum einen Beitrag in der Wertschöpfungskette – „value“ zu generieren und „Waste“ zu vermeiden.

Dadurch soll eine Kostenreduktion

- Steigerung der Effizienz
- Steigerung der Produktivität
- Steigerung der Qualität

erreicht werden.

Lean

Lean betrachtet also wie Dinge bearbeitet werden im Rahmen eines Kontinuierlichen und iterativen Verbesserungsprozesses (KVP)

Werden vorangegangene Punkte erreicht:

- Sinken die Kosten
- Erhöht sich die Kundenzufriedenheit
- Erhöht sich die Mitarbeiterzufriedenheit

Lean

Als Grundlage dienen 5 Prinzipien:

- Wert Spezifizieren (Customer driven Measurement)
- Wertschöpfung identifizieren und Waste entfernen
- Wertschöpfungsfluss (Do not be to early and definitivly do not be late)
- Mitarbeiter Einbinden und bevollmächtigen
- Kontinuierlicher Verbesserungsprozess

Lean

Identifizieren welcher Prozess(schritt) Wert generiert.

Jene die Wert generieren

=> beibehalten und KVP anwenden

Keinen Wert generieren aber notwendig sind

=> reduzieren

Keinen Wert generieren

=> versuchen zu eliminieren

Lean

Waste wird in 4 Typen definiert:

- Information (Redundanz, Medienbrüche,…)
- Prozesse (Defekte, Workarounts, Überprüfungen, Warten, Überproduktion, Hohe Lagerstände,…)
- Umwelt (Wegzeiten, Verplanen von ev. notwendigen Ressourcen,…)
- Personen (unklare rollen, Fehlendes Training, Unterbrechungen, Multitasking, (blockierende) Hierarchien & Strukturen,…)

Lean

Wertschöpfungsfluss

Badge processing vs. Single Item flow (lean flow)

Push vs. Pull

Lean vs. Trad. Management

Lean	trad. Management
Manamgent by sight	By status report (remote decisions)
Pull systems	Push systems
Reduce non-value added work	Speed up value added work
Single item flow	Badge production
Right process => right results	Results at all costs
Quality built in	Quality inspected
Ask questions	Give answers

Lean Zusammengefasst

Aus dem Lean Gedanken heraus gibt es verschiedene Methoden die mehr oder weniger große Teile aus dem Lean Gedankenset detaillierter beschreiben.

Bsp:

- Kaizen (KVP Grundgedanke)
- Jidoka („Menschliche“ Automatisierung)
- Gemba Walk
- Kata
- Kamishibai Board(rote/grüne Kärtchen)

Lean vs. agil

Lean Ursprung liegt in der Produktion, agil ist die Weiterentwicklung des Lean Gedankens mit dem Fokus

- Änderungswünsche des Kunden einbinden können
- Produkt schneller auf den Markt zu bringen

Anstatt (bei Lean)

- Möglichkeiten zu identifizieren um Qualität zu erhöhen
- Waste identifizieren um Kosten zu sparen

Lean vs. agil

Das Ziel von agile ist die Menge der nicht notwendigen Arbeit zu maximieren

Lean vs. agil

Also zwei unterschiedliche Sicht-/Herangehensweisen und unterschiedlicher Fokus

Lean => Waste reduzieren

Agile => schnelle Reaktion auf Änderungen

Lean => Genügsamkeit um Kosten zu reduzieren

Agile => Verluste vermeiden um Kosten zu reduzieren

Mittagspause

Agile

Alle agilen Vorgehensmodelle beruhen auf dem Agilen Manifest!

Agile manifesto

- ***Individuals and interactions*** *over processes and tools*
- ***Working software*** *over comprehensive documentation*
- ***Customer collaboration*** *over contract negotiation*
- ***Responding to change*** *over following a plan*

… while there is value in the items on the right,

we value the items on the left more.

Agile manifesto

- Die zentrale Idee liegt in der Befähigung auf Änderungen zu reagieren (bedingt eine neue (positive) Einstellung gegenüber Änderungen).

- Zusätzlich zum manifesto hat sich die Gruppe auf 12 Prinzipien geeinigt. (vgl. http://agilemanifesto.org/)

- Agile Methoden sind daher flexibel und lassen Raum für Veränderungen

- Mitarbeiter werden aufgefordert (in die Pflicht genommen) selbstständig zu denken und nicht nur Regeln zu befolgen. (Selbststeuernde Teams)

Höchste Priorität ist die fertige Lösung welche Anforderungen der Kunden erfüllt!

Was bedeutet Agile?

- Eine Sammlung von **Werten und Prinzipien …**
- Eine Sammlung von (besten) **Praktiken …**

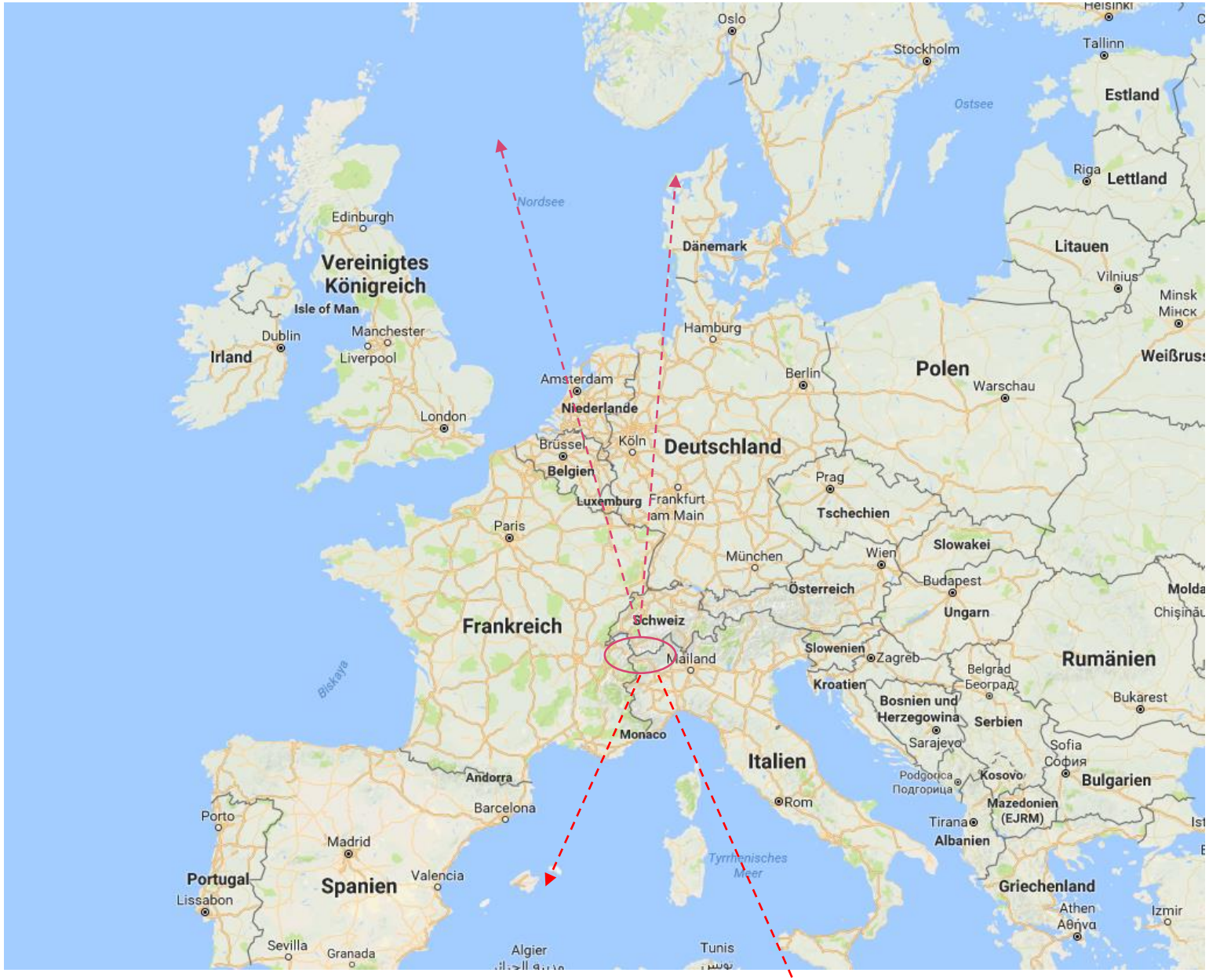
Bei denen das Team im Mittelpunkt steht

- Agilität ist eine Denkweise und ein (alternativer?) Ansatz um business value (Wertschöpfung) früher zum Kunden zu bringen

Was ist es nicht?

- Agility ist kein Allheilmittel
- … keine einfache Methode oder ein Werkzeugkasten
- … kein Garant für Erfolg
- … agile macht nicht immer Sinn
- …. Eine Reise ins Ungewisse

Keine Reise ins Ungewisse



Quelle: Google Maps

Die 12 Prinzipien

Die höchste Priorität besteht darin, den Kunden durch frühe und zahlreiche Lieferungen hochwertiger Software zufriedenzustellen.

Funktionierten Software muss regelmäßig geliefert werden. Je kürzer desto besser.

Agile Prozesse fördern eine kontinuierliche Entwicklung. Geldgeber, Entwickler und Benutzer sollten endlos ein beständiges Tempo beibehalten.

Vorteile:

- Kontinuierliche „Product increments“
- Kunde kann früher Ergebnisse nutzen
- Risiko des Kunden wird geringer
- Hoher Erkenntnisgewinn auf beiden Seiten

Die 12 Prinzipien

Veränderte Anforderungen sollen immer positiv aufgenommen werden. Sie stellen einen Wettbewerbsvorteil des Kunden dar

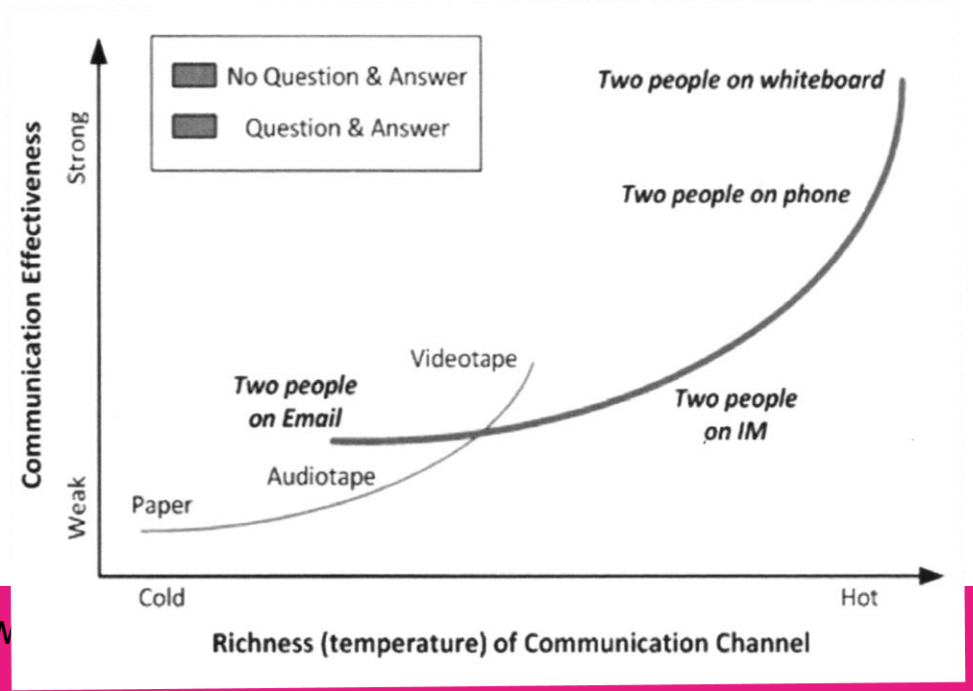
Anmerkung:

- Bedingt eine Änderung im Mindset im Team/Unternehmen als auch beim Kunden
- Für alle Vorteilhaft wenn Kunde das bekommt was er wirklich braucht

Die 12 Prinzipien

Mitarbeiter aus den Fachbereichen und der Entwicklung arbeiten täglich am Projekt zusammen

Die effizientesten und effektivsten Methoden zur Informationsübermittlung ist die direkte Kommunikation



Die 12 Prinzipien

Bauen sie Ihr Projekt um motivierte Mitarbeiter auf. Geben sie diesen MA die Umgebung und Unterstützung damit sie performant arbeiten können. Vertrauen sie darauf, dass sie Ihre Arbeit gut machen.

- Die besten Systeme entwickeln sich aus Teams die sich selbst organisieren.

Vorteile:

höheres Commitment

Team steuert sich selbst

Lösungsorientierung

Die 12 Prinzipien

- Ständige Aufmerksamkeit gegenüber technisch hervorragender Qualität verbessert die Agilität.

- Die Teams müssen in regelmäßigen abständen darüber reflektieren wie sie ihr Vorgehen weiter verbessern können und passen das Verhalten entsprechend an.

Vorteile:

- Im Model implementierter kontinuierlicher Verbesserungsprozess
- Awareness für Verbesserungspotential
- Team lernt (aus Fehlern)
- Hohes Qualitätsbewusstsein

Die 12 Prinzipien

Funktionierende Software ist der primäre Maßstab für den Fortschritt.

Einfachheit – die Kunst, die Menge der nicht geleisteten Arbeit zu Maximieren - ist essentiell

Vorteile:

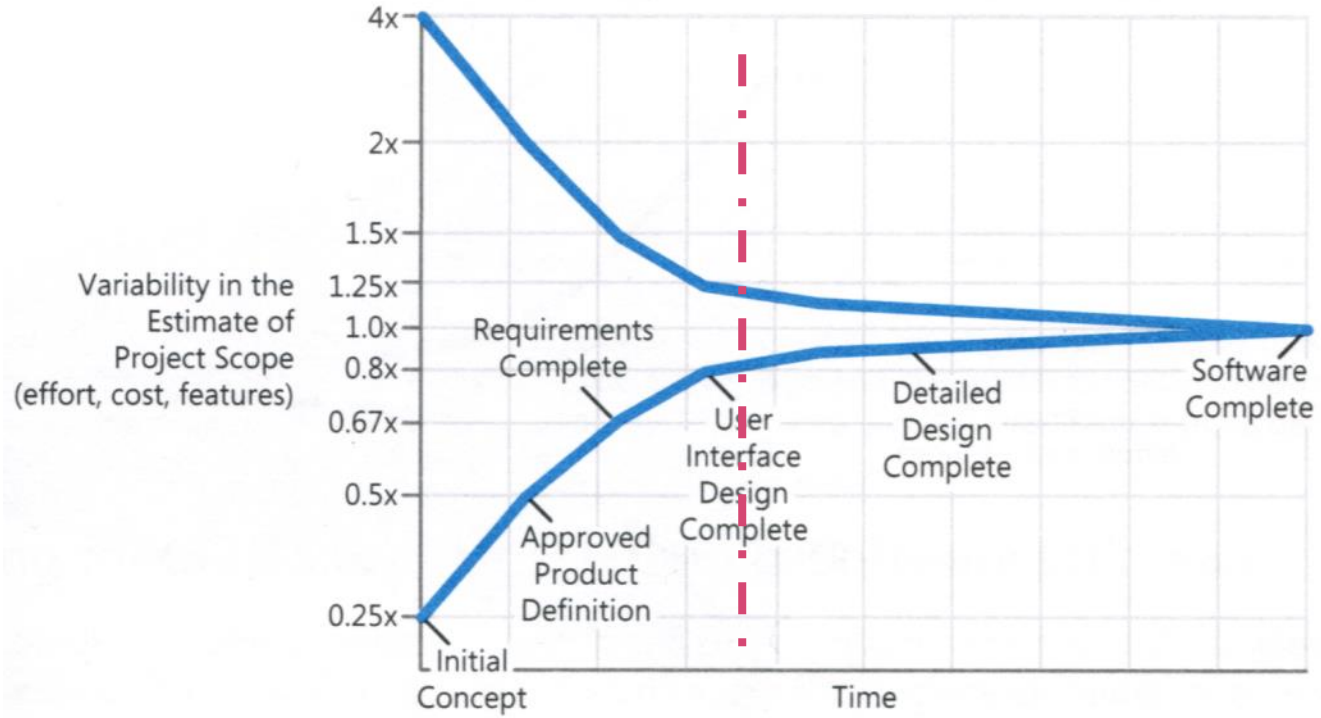
Kostengünstiger

Kunde bekommt das - und nur das – was er benötigt.

Müssen immer 100% der Anforderungen zu 100% spezifiziert werden?

- 80% des (Produkt)nutzens basieren auf 20% der Features (Powerpoint, Word,…☺)
- XML and GML (general markup language)
- 60% der Features von ausgelieferten Produkten werden selten bis sehr selten genutzt (Lacey 2011)

Funnel Kurve /Konus der Unsicherheit



- Zu Projektstart ist die Unsicherheit in Bezug auf die Anforderungen 16 mal so hoch wie am Ende des Projektes (Boehm).
- Nach 30% des Projektlaufzeit hat sich diese Unsicherheit von 16x auf 1,5 x reduziert.

Zusammenfassung

Selbststeuernde Teams übernehmen Verantwortung

Regelmäßige Lieferungen

Einbindung des Kunden

Übersichtliches Reporting

Im Prozess implementiertes Controlling

Ausreden warum es nicht funktioniert

Unser Produkt ist zu kompliziert
Um nicht alles im vorhinein zu planen

Unsere Qualitätsansprüche sind zu hoch
Um keine Dokumentation und keinem Plan zu folgen

Hier wurde Agile Falsch verstanden!

Wir haben bereits viel Geld in fixe Maschinen und Werkzeuge investiert

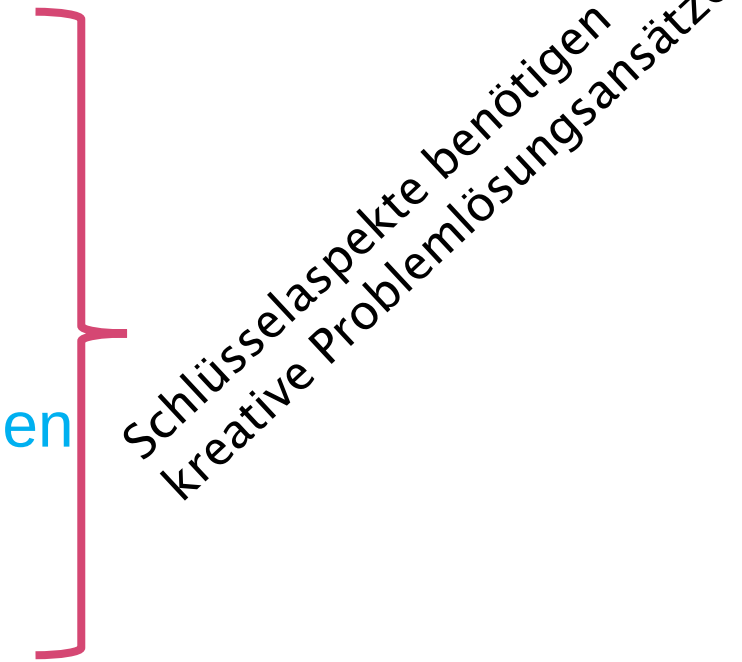
Unser Produktdesign ist zu verwoben um iterative Module zu entwickeln

Diese Hindernisse können iterativ beseitigt werden

Ausreden warum es nicht funktioniert

Unsere Kunden sind nicht agil genug
um diesen Ansatz zu folgen

Schlüsselschritte des Prozesses benötigen
lange Vorlaufzeiten und können nicht in
Sprints gegossen werden



Pause

Case Study

Hausbau eines Einfamilienhauses mit agilem Vorgehensmodel

Welche Rahmenbedingungen sind notwendig?

Nach was könnte priorisiert werden?

Case Study

Hausbau eines Einfamilienhauses

Pläne:
Einreichplan
Polierpläne
Energieausweis

Keller,
EG
OG

Case Study

Hausbau eines Einfamilienhauses.

Phasen:

Planung
Vorbereitung
Rohbau
- Keller
- EG
- OG

Dach
- Dachstuhl
- Dachdeckung

Innenausbau

Case Study

Hausbau eines Einfamilienhauses.

Wie könnten anhand der Phasen die Iterationen geplant werden?